

IN THE CLAIMS

Please amend the claims as follows:

1. (original) A method for encrypting a digital data stream in a transmission system that uses orthogonal codes for the modulation, wherein - a  $k^{\text{th}}$  transmitter constructs a  $k^{\text{th}}$  connection for the  $k^{\text{th}}$  digital data stream ( $d^{(k)}$ ), - for the encryption, the digital data stream ( $d^{(k)}$ ) of the transmitter is mixed with a spreading code that is assigned to this  $k^{\text{th}}$  connection, - different spreading codes ( $g_1^{(k)}, g_2^{(k)} \dots g_H^{(k)}$ ) from a defined set ( $G_i$ ) are assigned and - through the mixing a transmission signal ( $s^{(k)}$ ) is produced, characterized in that the degree of encryption of the  $k^{\text{th}}$  digital data stream ( $d^{(k)}$ ) is increased during the  $k^{\text{th}}$  connection through the allocation of • a sequence for the application of the different spreading codes ( $g_1^{(k)}, g_2^{(k)} \dots g_H^{(k)}$ ) and/or a hop interval ( $I_{\text{hop}}$ ).
2. (original) A method as claimed in claim 1, characterized in that a permutation function ( $S_i$ ) defines the sequence of the application of the content of a set of spreading codes ( $G_i$ ) by stating the position ( $\{p_1, p_2 \dots p_M\}$ ).
3. (original) A method for encrypting a digital data stream that is to be transmitted, wherein after the connection set-up,

necessary parameters for the transmission and recovery are transmitted, characterized by the steps:

- communication of an encryption key (200) and thus:
  - establishment (210) of a permutation function ( $S_i$ ),
  - establishment (220) of a set ( $G_i$ ) of spreading codes, and/or
  - establishment (230) of a hop interval ( $I_{hop}$ ),

wherein the last three steps mentioned (210, 220, 230) can be carried out in any order.

4. (original) A method for encrypting a digital data stream, characterized by the execution of a first permutation procedure (400) which contains a loop with the following steps:

- setting (410) of an interval (n) to "1";
- waiting (420) for the end of a predefined hop interval ( $I_{hop}$ );
- increasing (430) the interval (n) by the value 1;
- carrying out a comparison (440) to see whether the current value of the interval (n) is greater than the total number (M) of the elements of a permutation function ( $S_i$ ) which states the positions of the spreading code ( $g_n$ ) of a set ( $G_i$ ) of spreading codes that is to be used for encrypting the digital data stream, wherein alternatively the following takes place:

- if the comparison has a positive result: resetting of the interval (n) to "1";.

- if the comparison has a negative result: equating the current spreading code ( $g_n$ ) with the spreading code ( $g_{p_n}$ ) that stands at the position ( $p_n$ ) stipulated by the permutation function ( $S_i$ ).

5. (currently amended) A device (1) for carrying out a method as claimed in ~~any one of the preceding claims~~claim 1, characterized in that the device has a first code generator (2) that creates the respectively current spreading code ( $g_n$ ).

6. (original) A method for decoding a received digital data stream that was sent encrypted, characterized by the execution of a second permutation procedure (800) that contains a loop with the following steps:

- setting (810) an interval ( $n$ ) to "1";
- waiting (820) for the end of a predefined hop interval ( $I_{hop}$ );
- increasing (830) the interval ( $n$ ) by the value 1;
- carrying out a comparison (840) to see whether the current value of the interval ( $n$ ) is greater than the total number ( $M$ ) of the elements of a permutation function ( $S_i$ ) which states the positions of the spreading code ( $g_n$ ) of a set ( $G_i$ ) of spreading codes that is to be used for decoding the encrypted digital data stream, wherein alternatively the following takes place:

- if the comparison has a positive result: resetting of the interval (n) to "1";

- if the comparison has a negative result: equating the current spreading code ( $g_n$ ) with the spreading code ( $g_{p_n}$ ) that stands at the position ( $p_n$ ) stipulated by the permutation function ( $S_i$ ).

7. (original) A device (3) for carrying out a method as claimed in claim 6, characterized in that the device (3) has a second code generator (4) that produces the current spreading code ( $g_n$ ).

8. (currently amended) A transmission system that uses orthogonal codes for the modulation, with a device for encrypting a digital data stream, in particular a device (1) as claimed in claim 5, wherein the digital data stream ( $d^{(k)}$ ) is mixed with a spreading code, and with a device for decoding a digital data stream that was ~~transmitted encrypted, in particular a device (3) as claimed in~~ claim 6 sent encrypted, characterized by the execution of a second permutation procedure (800) that contains a loop with the following steps:

- setting (810) an interval (n) to "1";
- waiting (820) for the end of a predefined hop interval ( $I_{hop}$ );
- increasing (830) the interval (n) by the value 1;

- carrying out a comparison (840) to see whether the current value of the interval (n) is greater than the total number (M) of the elements of a permutation function ( $S_i$ ) which states the positions of the spreading code ( $g_n$ ) of a set ( $G_i$ ) of spreading codes that is to be used for decoding the encrypted digital data stream, wherein alternatively the following takes place:

- if the comparison has a positive result: resetting of the interval (n) to "1";

- if the comparison has a negative result: equating the current spreading code ( $g_n$ ) with the spreading code ( $g_{p_n}$ ) that stands at the position (p n) stipulated by the permutation function ( $S_i$ ), characterized in that it has means for

- carrying out encryption,
- carrying out decoding of a digital data stream that was transmitted encrypted.

9. (original) Use of one of the methods mentioned above in a cordless or line-based network.